



# A fogalmi gráfok

## Conceptual Graphs for a Data Base Interface

---

John F. Sowa

1976

IBM Journal of Research and Development, 20(4), pp. 336–357.

- formalizmus, fogalmi gráfok (*conceptual graphs*)
- cél: adatok leírása a felhasználó szempontjából, adatok elérése a rendszer szempontjából = **interfész**
- extra cél: funkcionális függőségek reprezentációja, nem explicit következtetések

# Bevezetés

---

- adatbázis-rendszerek: elérési módszerek, általánosítva
  - fő kérdés: hogyan tároljuk, hogyan érjük el az adatokat
  - ami kimarad: mit jelent ez az adatbázis az azt használó emberek számára
- "szakadék" – ezt hidaljuk át a fogalmi gráfokkal:
  - közvetítő szerep ember és gép között
  - ember **kérdez**, ezt a rendszer **lefordítja** fogalmi gráffá, aztán **keres** kapcsolódó gráfokat az adatbázist leíró gráfok között, ha **talál**, akkor azzal **eléri** az adatokat, és **kiszámolja** a választ
- nem adattárolás, hanem adat- és kölcsönöskapcsolat-leírás
- három fontos előny (mint formális leírás):
  1. (direkt leképezés támogatása a relációs adatbázisra)
  2. (természetes nyelv szemantikai bázisaként használhatók)
  3. automatikus következtetések támogatása, hogy nem explicit kapcsolatokat is kiszámolhassunk: ez a lényeg itt; nem csak a logikai kapcsolatok reprezentációja a cél, hanem hogy olyan következtetéseket tudjon a rendszer csinálni, amikkel az eredeti kérdés megválaszolható

HIRE	EMPLOYEE	MANAGER	DATE
	Robert Lee	John Brown	3/1/74
	Mary Smith	John Brown	9/7/70
	Tom Jones	Mary Smith	8/8/75

**Figure 1** The HIRE relation.

- reláció: HIRE
- domainek: EMPLOYEE, MANAGER, DATE
- alatta n-tuple-ök, amikre a reláció igaz
- Hogyan interpretáljuk? Több lehetséges válasz, pl. az adott dolgozó az adott menedzser által az adott dátumon lett felvéve; de pl. lehet az adott menedzser kinevezésének időpontja és az első dolgozó neve, akit ő felvett stb.
- a reláció jelentése: **intenzió**
- az n-esek halmaza: **extenzió**

Három fontos intenzionális információ:

1. funkcionális függőségek

- melyik domain lehet *kulcs*
- melyik az, ami a kulcstól függ
- példában? (Dolgozó a kulcs, mindegyikhez van egy őt felvevő menedzser és a felvétel dátuma)

2. domainszerepek

- domaineink közti kapcsolatok

3. domaineinkre vonatkozó megszorítások

- a megengedett értékek

Nem elég, hogy reprezentáljuk az intenziókat, az új infók elfogadhatóságát is kell tudnunk ellenőrizni velük; Fogalmi gráfok: intenzionális formalizmus, a következő feltételeknek akar megfelelni:

1. familiáris **konvenciók** (*familiar conventions*): kérdezhessek a számítógépes rendszer ismerete nélkül
2. **automatikus következtetések** (*automatic inference*): olyan kapcsolatokra tudjon következtetni a rendszer, amik nincsenek explicit módon tárolva az adatbázisban
3. **természetesség** (*naturalness*): lehető legközelebb álljon a formalizmus a természetes nyelv szemantikájához
4. **szemantikai integritás** (*semantic integrity*): pontos visszatükrözése legyen az adatbázis a való világnak a megszorítások segítségével
5. (gyorsaság, megbízhatóság)

Ennek tükrében a fogalmi gráfok:

- fogalmak és a domainszerepeket leíró fogalmi kapcsolatok hálózata
- a fogalmi sémák (*conceptual schemata*) és a működési gráfok (*working graphs*): egymásba ágyazott funkcionális függőségek hálózata, ez az adatbázisra van leképezve
- a válaszolás: a rendszer összeállít egy működő gráfot, amelyben a megfelelő domainszerepek és funkcionális függőségek vannak



# A fogalmi gráfok

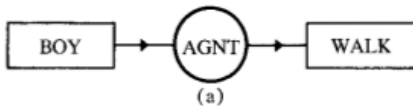
---

## Alapfogalom: **fogalom** (*concept*)

- doboz, típuscímkével (*sort label*)
- címke nagybetűs szó
- szimbólum, bármit reprezentálhat, amire gondolhatunk: entitás, matematikai függvény, egy absztrakció, akármi
- a világnak az adatbázisban rögzített aspektusaival kapcsolatban legalább domainenként egy fogalom definiálva kell legyen
- van általánosabb (PERSON), és kevésbé általános (EMPLOYEE)
- az általánosság kifejezésére ezek rendezve vannak: **altípusok** (*subsort*) (EMPLOYEE - PERSON)
- azoknak a dolgoknak a halmaza, amikre  $a$  áll, részhalmaza azoknak a dolgoknak a halmazának, amikre  $b$  áll, ha  $a$  subsort-ja  $b$ -nek
- fogalmak akkor is disztinktek, ha ugyanazt reprezentálják (ha minden PERSON történetesen EMPLOYEE)
- **közös altípus** (*common subsort*): LION, TIGER, JAGUAR - FELINE, WILD ANIMAL

## Fogalmi kapcsolatok (*conceptual relations*)

- címkézett körök, egy vagy több linkkel (itt a példák mindig diádikus relációk lesznek)
- élek számozva, rajta nyíl az irányt mutatja



## Fogalmi gráf (*conceptual graph*)

- *véges*
- *összefüggő* (eljuthatok bármely A-ból bármely B-be)
- *irányítatlan*
- *páros* (két jól körülhatárolható halmaz, halmazon belül nincs kapcsolat - conceptek és relációk)

gráf, két típusú node-dal: fogalmak és relációk

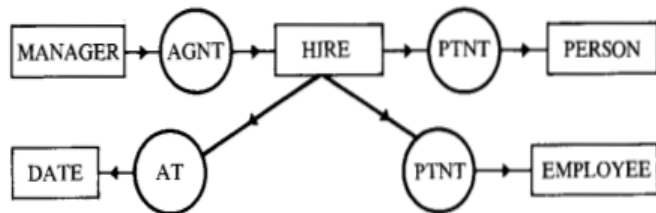
# A formázási szabályok

---

## Jólformált kombinációkhoz formációs szabályok

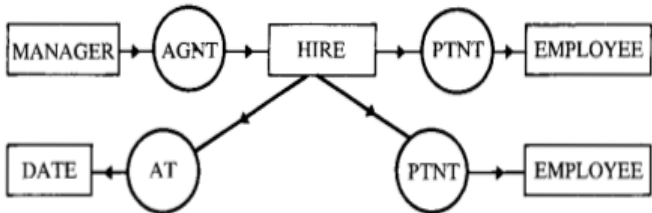
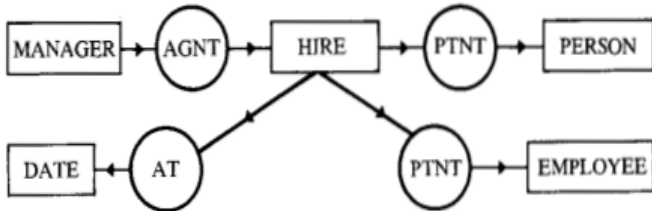
- Minden gráf, ami egyetlen fogalmat tartalmaz, jólformált.
- Minden más jólformált gráf az alábbi szabályok (ismétlődő) alkalmazásával jön létre:
  1. **Másolás** (*copy*): bármely jólformált gráf pontos másolata jólformált.
  2. **Leválasztás** (*detach*): ha egy jólformált gráfból elveszünk egy fogalmi relációt, akkor a fennmaradó (összefüggő!) gráf jólformált
  3. **Megszorítás** (*restrict*): ha egy  $a$  fogalmat annak subsortjával helyettesítünk, a jólformált gráf jólformált marad.
  4. **Egyesítés** (*join*): ha ugyanazon címkéjű fogalom szerepel egyik gráfban és másik gráfban, akkor a fogalomnak az egyik gráfból való törlésével és az éleinek a fogalom másik gráfban szereplő előfordulásához kötésével jólformált gráfot kapunk.
- Feltétel: jólformált gráfok kiinduló halmaza

A következő diákon találjuk ki, hogy a felső gráfból melyik formázással keletkezett az első. (Válasz mindig a következő dián.)

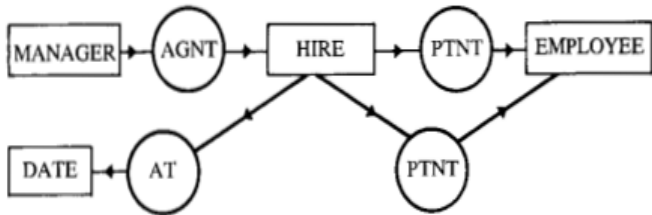
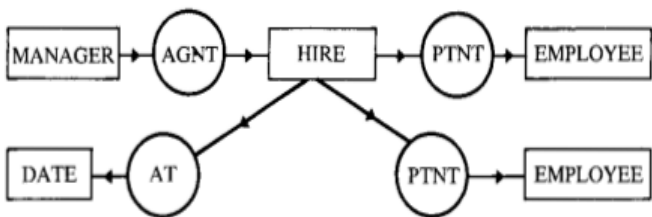




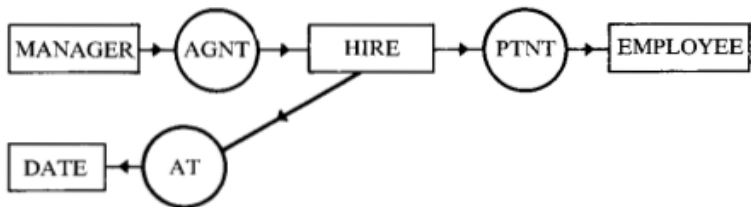
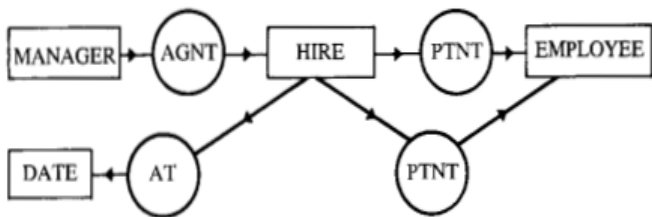
Ez volt a JOIN.



Ez volt a RESTRICT.

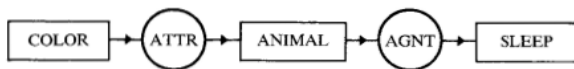
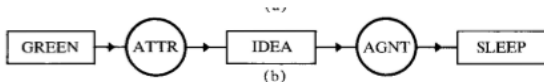


Ez megint egy join.



Ez pedig egy detach.

- formázási szabályok != következtetési szabályok (első gráf nem oké, ezt a második és a harmadik, egyenként jólformált gráfból megszorítással és *join*-nal létrehozott negyedik gráf mutatja, mert abból nem jöhet létre sehogy az első)





# Származtatott képzési szabályok

---

Lényeg: egyszerűsítés, több lépés egyszerre

## Projekció

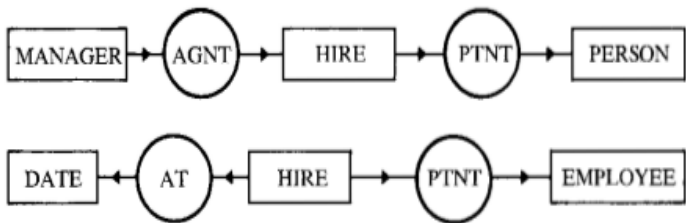
- $n$  db leválasztás és  $n$  db megszorítás, 0 join (tetszőleges sorrend)
- méret csökken
- lehet, hogy egy összefüggő helyett több kisebb gráf, kevesebb elemmel
- a megszorítás pedig specifikusabbá teszi
- egy fogalmi gráf projekciója mindig rávetíthető az eredeti gráf egy részgráfjára, legfeljebb a fogalmak címkéje lesz specifikusabb; ez a részgráf az új gráf **projektív eredete** (*projective origin*)
- **közös projekció**: ha  $u$  a projekciója  $v$ -nek és  $u$  a projekciója  $w$ -nek, akkor  $u$  a közös projekciója  $v$ -nek és  $w$ -nek  $\rightarrow$  izomorf lesz a két projektív eredet

Két gráf egyesítése a közös projekciójuk mentén: a projekció két projektív eredete fektethető egymásra, már csak a fogalmakat kell helyettesíteni benne a közös subsorttal

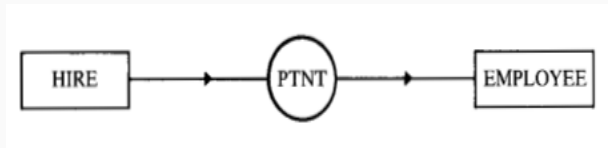
- A közös projekció eredeti gráfbeli projektív eredetének minden fogalma és relációja **le lesz fedve a joinnal** (*covered by the join*)
- ez (a fedés) a kérdésre adott válasz miatt fontos: lényegében olyan válaszgráfot kell generálnia a rendszernek a kérdésgráfra, amely két gráfot ha joinoljuk, a join lefedti az egész kérdésgráfot

## A maximális közös projekció felé:

- **kernel:** a közös projekció kernele egy tetszőleges fogalom a közös projekcióban, és ennek a két megfelelője a két projektív eredetben (a három egymásra pakolt fogalom)
- **maximális közös projekció:** ha nincs más gráf, ami ugyanezzel a kernellel lenne közös projekció, úgy, hogy ez a maximális közös projekció is projekciója annak a másiknak
- **maximális join:** ha a maximális közös projekción join-olunk



A maximális közös projekció:



Feladat: hogy néz ki akkor a join ezen a maximális közös projekción?

Kérdés: más maximális közös projekció? (következő dián a válasz)

Igen, lehetne más maximális közös projekció: a MANAGER fogalom, és az első gráf MANAGER, a második gráf EMPLOYEE fogalma lenne a kernel.

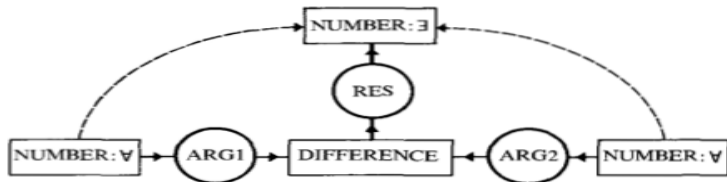
# Értékek és kvantorok

---

Eddig általánosságban beszéltünk, nyilván ennél pontosabbnak kell lennünk: fogalmak társítása konkrét példányokkal, azaz értékeket kell rendelnünk hozzájuk (NUMBER:1234, CAT: Morris). Egy fogalom lehet:

- indefinit: csak a címke van a dobozban
- konstans: specifikus érték van a dobozban
- kvantoros: logikai kvantor:  $\forall$  és  $\exists$ , illetve  $\exists!$  és  $\exists$  – set ('there exists a set of sorts...'), a hatókör jelzése szaggatott vonallal

Kérdés: mit ábrázolhat az alábbi gráf?





A *különbség* definíciója: minden  $x$ -re és  $y$ -ra létezik olyan  $z$ , amelyre  $x - y = z$  igaz, valahogy így.

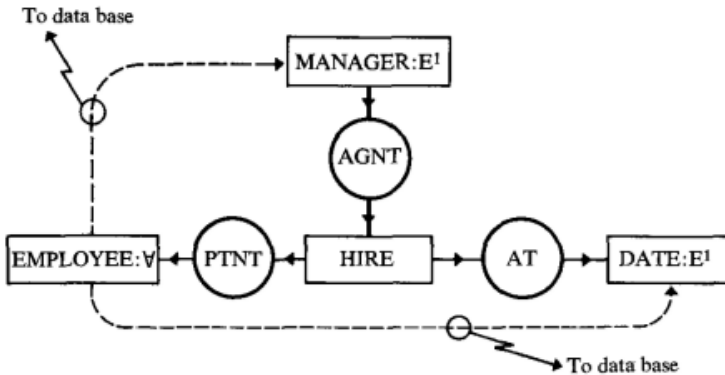
- a szaggatott vonalak igazából függvények: **funkcionális függőségek** (*functional dependencies*):
  - funkcióélek (*function links*) halmaza
  - ezek az élek egy vagy több forrásfogalomtól egy célfogalomba tartanak
  - minden funkcionális függőséghez tartozik egy **elérési eljárás** (*access procedure*): ha minden forrásnak van értéke, akkor az eljárás ki tudja számolni a cél értékét
- két függvény: *value()* és *quant()*
  - bármely fogalomra
  - ha mindkettő meghatározatlan, akkor a fogalom is indefinit
  - ha a *value()*-nak van értéke, akkor a fogalom egy konstans
  - ha a *quant()*-nak van értéke, akkor az a négy logikai kvantor egyike

# Fogalmi séma

---

A **fogalmi séma** egy olyan fogalmi gráf, amiben már vannak kvantorok és funkcionális függőségek; átmenet a fogalmi gráfok és maga az adatbázis között. Egy kérdés megválaszolásához a rendszer generál (vagy talál) egy olyan sémát, aminek a funkcionális függőségei megadják a választ. A válasz lehet egy skaláris ( $\exists!$ ), vagy halmazok ( $\exists - set$ )

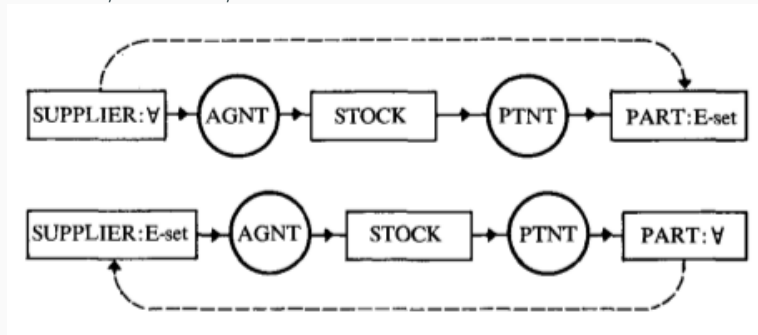
- egy vagy több funkcionális függőség
- ha  $c$  forrás, de nem cél:  $quant(c) = \forall$
- ha  $c$  forrás is és cél is:  $quant(c) = \exists!$
- ha  $c$  csak cél:  $quant(c) = \exists! \text{ or } \exists - set$
- minden más fogalom indefinit, és *szelektornak* hívjuk
- egy fogalom csak egy függőségnek lehet a célja



Key domain: EMPLOYEE, szelektor: HIRE, nonkey domain: MANAGER, DATE.

Egy relációnak több kulcsdomain-halmaza is lehet: minden lehetséges kulcsdomain-halmazhoz külön séma kell. Példa:  $difference(x, y, z)$ , ha nem  $x$  és  $y$  az ismert, hanem pl.  $y$  és  $z$ .

Alábbi gráfok: STOCK reláció, many-to-many, minden SUPPLIER-re van sok PARTS, amit ő *stockol*, és fordítva is. Ezt egy gráfon nem lehet ábrázolni, kell kettő, és a SUPPLIER és a PARTS is kulcs lesz.



# Egy lekérdezés megválaszolása

---

- egy tipikus lekérdezés:

Találj meg minden olyan  $(x, y)$  párt, amelyre  $R(x, y, a, b, c)$  igaz.

- ha  $R$  az alaprelációk egyike, akkor triviális a válaszadás
- ha viszont nem az alaprelációk egyike, akkor relációk kombinációjával kell meghatározni
- (több meglévő rendszer bemutatása szoftveres szemmel, kicsit felhasználóival is)
- fogalmi gráfokkal: sémákat joinolunk
  - először a lekérdezés gráfja
  - utána meghatározzuk, melyik sémákkal joinolunk
  - utána megnézzük, milyen adatbázis-elérések kellenek, hogy kiszámoljuk a választ



Mi beütünk egy kérdést, majd:

- a bemeneti elemző egy jólformált fogalmi gráfot készít belőle ( $q$ )
- minden olyan fogalomnak, aminek az értékét meg akarjuk határozni, most egy kérdőjelet adunk
- a kérdőjeles fogalmak értékének meghatározásához a rendszer generál egy válaszgráfot ( $w$ ), amely megfelel annak, hogy:
  1.  $w$  egy jólformált fogalmi gráf
  2.  $w$  igaz, ha az adatbázis helyes
  3. az egész  $q$ -t lefedi annak  $w$ -vel való joinolása (=  $w$  magában foglalja a lekérdezési gráf minden domainszerepét és relációját, noha lehetnek ezek tovább szűkítve)
  4. minden olyan fogalomra, aminek  $q$ -ban van értéke, a  $w$ -beli megfelelőben ugyanaz az érték van (=ugyanazokról az entitásokról beszélünk)
  5. minden olyan fogalomra, ami  $q$ -ban kérdőjeles, a  $w$ -beli megfelelőben van érték
- ha ez alapján nincs meg, akkor visszakérdez

# Algoritmusok a válaszgráf generálására

---

*Who hired Lee?*

Feladat: megrajzolni ezt a lekérdezés-gráfot.



össze a 30-as számú dián látható gráffal!

- Lee jön az univerzális kvantor helyére (EMPLOYEE Lee)
- ha egy függvény célja van kérdőjelezve, nincs gond, kiszámoljuk: a MANAGER funkcionálisan függ az EMPLOYEE-tól: adatbázisból kihúzzuk a nevét
- mi volt a gond? ha a kérdőjeles fogalom nem célja egyetlen függvénynek sem; megoldás a join volt, egy olyan sémára, ahol az adott fogalom célfogalom.

Általános algoritmus (A):

Indulj a kérdőjeles fogalmaktól; joinold a lekérdezés gráfját olyan fogalmi sémákkal, ahol a kérdőjeles fogalmakat célfogalmak fedik; propagáld a kérdőjeleket a funkcionális kapcsolatokon visszafelé, számold ki az értékeket ott, ahol a forrásfogalmaknak van értéke; addig ismételd, míg meg nem válaszolod a kérdést.

- bajok: ha pl. az eredeti kérdés hiányos, sosem fogok a végére érni
- B algoritmus:
  - Minden egyes új séma és a fejlődő válaszgráf joinolása le kell fedjen legalább egy fogalmat a kérdezési gráfból.
- ezzel megakadályozzuk, hogy túl nagyra nőjön; a 'Who was the person who hired the person who hired the person who hired Jones?' meglesz neki; a helyes válasz viszont még nincs garantálva

## A sematikus join:

- két fogalmi gráf joinolása, vagy egy fogalmi gráf önmagával való joinolása, a következő feltételekkel:
- maximális
- minden funkcionális él tovább öröklődik
- ha egy indefinit fogalmat kapcsolunk egy kvantorshoz, az eredmény ugyanazt a kvantort fogja tartalmazni
- ha az  $\exists!$ -t és az  $\forall$  kvantort joinolom, az eredményben az  $\exists!$  lesz
- ha az  $\forall$ -t és a  $\forall$ -t joinolom, az eredményben is az lesz
- a kvantorok bármely más joinolása tiltott

A gráf, ami egy sematikus join eredménye, fogalmi séma lesz.



A fogalmi sémák  $S$  halmazára által meghatározott **sematikus univerzum** minden séma halmazára, amely a következő műveletekkel áll elő:

- minden  $S$ -beli séma a sematikus univerzumban van
- minden  $s$  séma önmagával való sematikus join-olásának eredménye is a sematikus univerzumban lesz
- ugyanez két külön sémára

## A válaszgráf:

- egy fogalmi gráf, amelyet úgy nyerünk, hogy az  $s$  séma minden kvantoros fogalmához értéket rendelünk, úgy, hogy:
  - ahol  $\forall$  van, ott a  $permissible(sort(c))$  halmazból választunk értéket
  - ha a forrásfogalmaknak van értéke, a célfogalomnak nincs, számolj
  - ha marad érték nélkül célfogalom, vedd el a gráfot

## A lekérdezési gráf:

- jólformált
- nincs funkcionális függőség, nincs kvantor
- minden fogalomnak értéke vagy kérdőjele van

*A working graph:*

- a válaszhoz vezető út lépései...
- az aktuális gráf és valamelyik séma (vagy önmaga) joinjai

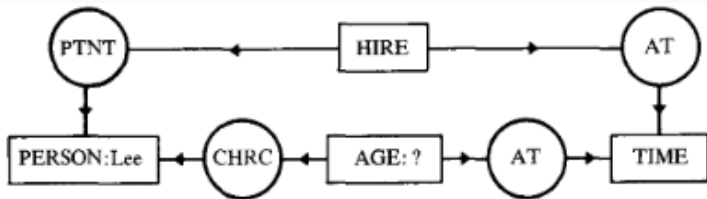
## Preferencia pontszámok:

- egy join értékelésére
- 1 pont minden fogalomért a lekérdezési gráfban, amit a join lefed
- 1 pont minden fogalomért a *working graph*-ban, amit a join lefed
- ha egy kérdőjeles fogalom a *working graph*-ból egy célfogalommal van fedve a joinban, 1 pont; ha forrásfogalommal, -1
- ha egy *working graph*-beli, értékkel bíró fogalom a joinban forrásfogalommal van fedve, 1 pont; ha célfogalommal, -1
- minden olyan fogalomért és relációért a lekérdezési gráfból, ami még nem volt sosem fedve, de most van, 1 pont

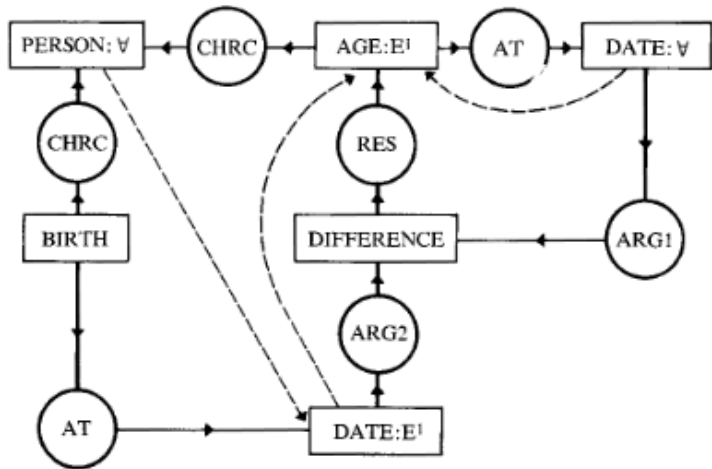
Példa

---

'What was Lee's age when hired?'



- az AGE az első: keresünk egy olyan sémát, ahol funkcionálisan függ valami mástól, ami kiszámolható





A maximális közös projekció:



A pontszámok:

- 3 pont a 3 lefedett fogalomért
- 3 pont, mert mindhárom benne van a lekérdezési gráfban
- 1 pont, mert egy kérdőjeles fogalom le van fedve egy célfogalommal
- 1 pont, mert egy értékkel rendelkező fogalom le van fedve egy forrásfogalommal
- 5 extra pont, mert 5 olyan fogalom és reláció van itt lefedve, ami korábban még nem volt → 13 pont

A sematikus join:

